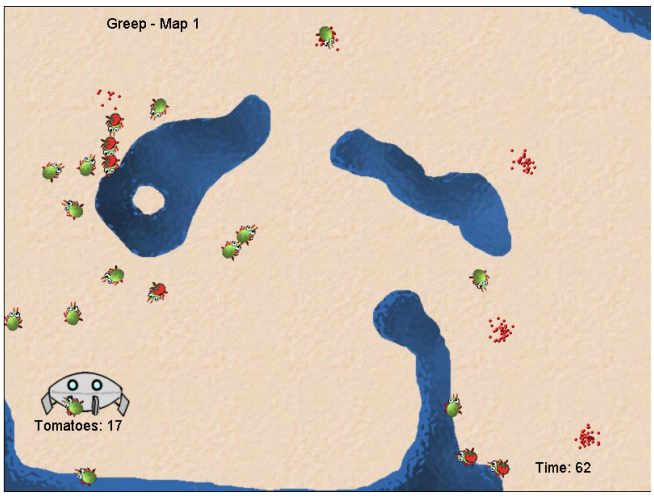
Greeps

The BHS computer science programming competition

The Greeps have come to Earth! And they like tomatoes. As soon as Greeps have landed, they walk all over the place, always looking for tomato deposits. But time is running out . . .

Welcome to the BHS computer science programming competition. Write your own `Greep` class and win the top honors!

In this competition, your task is to help the Greeps in their quest to collect as many tomatoes in a limited amount of time. You are to modify the provided `Greep` class according to specific rules. You will then submit your modified `Greep` class to enter the competition.



Tomatoes. Greeps love tomatoes. They eat tomatoes. In fact, they eat nothing else but tomatoes. Since there is an acute tomato shortage on their home planet they have to collect as many as they can on Earth.

You can only edit the `Greep` class. You cannot change any of the other classes. Program some intelligence into the little critters, and make them carry the tomatoes to their spaceship quicker than ever.

Some tips

You will see that `Greep` is a subclass of `Creature` and `Actor`. You can use any of the inherited methods (use of some `Actor` methods are restricted – see Rules 5 and 7 on the next page).

Greeps can only communicate by spitting paint drops (in three colors!) onto the ground. These paint drops can serve as markers to convey messages to other Greeps. Greeps do not communicate directly!

Greeps cannot load tomatoes on their own – a Greep can only load a tomato onto another Greep.

Greeps have some limited memory: one byte and two Boolean flags. You can use this for whatever you like, but you cannot extend it.

Also, the only part of the World that Greeps can see is their immediate location.

You need to have Greenfoot installed to run the competition. Once Greenfoot is installed, download, extract, and save the Greeps project to your computer. You can open the scenario in Greenfoot by clicking on the 'project' file. Then modify the code in the `Greep` class using the following rules:

Greeps Competition Rules

Rule 1: Only change the `Greep` class. No other classes may be modified or created. Indeed, you will only be submitting the `Greep` class for the competition and nothing else.

Rule 2: No additional variables. You cannot extend the Greeps' memory. That is: You are not allowed to add any instance variables to the class. You can use the provided one-byte memory, however.

Rule 3: Because Greeps can only make one step at a time, they cannot move more than once per '`act`' round. No matter how many `move` statements you write in your code, only ***one*** `move` statement will be executed per '`act`' round. In other words, there is no point in adding multiple `move` statements to your code, because they will be ignored.

Rule 4: Greeps cannot communicate directly with other Greeps. No variable access or method calls to other Greep objects are allowed. (Greeps can communicate indirectly via the paint spots on the ground, however.)

Rule 5: Greeps are extremely nearsighted. They do not have any distant vision. Greeps can only look at the world at their immediate location. Greeps are almost blind and cannot look any further.

Rule 6: No creation of objects. You cannot create scenario objects (instances of user-defined classes, such as `Greep` or `Paint`). Greeps have no magic powers - they cannot create something out of nothing.

Rule 7: No teleporting. You can not use methods from the `Actor` class that cheat normal movement (such as `setLocation`). No moving of tomato piles or ships.

Rule 8: No hard coding of maps and no altering of map objects. No changing of images.

Good Luck! You have three different Worlds in which to try to come up with the winning strategy. Make sure to run your simulation multiple times with the same code to determine if the numbers you are getting are consistent or just due to random chance. You should consider the following scale based on your final score in the initial 3 Worlds to gauge your success.

0-10 = pwned 10-30 = noob 30-50 = leet 50-70 = haxor >70 = pwnage

But wait; there's more! There actually are a series of 10 Worlds that you will be competing with, but only the first three are known to you in the beginning. Sometime before the actual competition, you will have a chance to run your best effort on all 10 Worlds to see how you really stack up. So, think of programming logic that will work on many different World layouts. When you are ready to turn in your finished work, navigate to the folder where your code is located, and find your `Greep.java` file there. Compress (Control-click on the file name and then select **Compress' greep.java**') and submit that file (just `Greep.java.zip` and nothing else). I will then download your file and run it on my computer.